



```
CCCCCCCC 000000 BBBB BBBB 111111 NN NN VV VV UU UU SSSSSSSS EEEEEEEEE
CCCCCCCC 000000 BBBB BBBB 111111 NN NN VV VV UU UU SSSSSSSS EEEEEEEEE
CC        00    00 BB      BB 11    NN NN VV VV UU UU SS      EEEEEEEEE
CC        00    00 BB      BB 11    NN NN VV VV UU UU SS      EEEEEEEEE
CC        00    00 BB      BB 11    NNNN NN VV VV UU UU SS      EEEEEEEEE
CC        00    00 BB      BB 11    NNNN NN VV VV UU UU SS      EEEEEEEEE
CC        00    00 BBBB BBBB 11    NN NN VV VV UU UU SSSSSS EEEEEEEEE
CC        00    00 BBBB BBBB 11    NN NN VV VV UU UU SSSSSS EEEEEEEEE
CC        00    00 BB      BB 11    NN NN VV VV UU UU SS      EEEEEEEEE
CC        00    00 BB      BB 11    NN NN VV VV UU UU SS      EEEEEEEEE
CC        00    00 BB      BB 11    NN NN VV VV UU UU SS      EEEEEEEEE
CC        00    00 BB      BB 11    NN NN VV VV UU UU SS      EEEEEEEEE
CCCCCCCC 000000 BBBB BBBB 111111 NN NN VV VV UU UU SSSSSSSS EEEEEEEEE
CCCCCCCC 000000 BBBB BBBB 111111 NN NN VV VV UU UU SSSSSSSS EEEEEEEEE

LL        111111 SSSSSSSS
LL        111111 SSSSSSSS
LL        11    SS
LL        11    SS
LL        11    SS
LL        11    SS
LL        11    SSSSSS
LL        11    SSSSSS
LL        11    SS
LL        11    SS
LL        11    SS
LL        11    SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS
```

(2) 48  
(3) 60  
(4) 95

HISTORY ; Detailed Current Edit History  
DECLARATIONS  
COB\$\$\$INVOKE\_USE

```
0000 1      .TITLE COB$$$INVOKE_USE COBOL Invoke USE Procedure
0000 2      .IDENT /1-004/ ; File: COBINVUSE.MAR
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28 FACILITY: COBOL SUPPORT
0000 29 ++
0000 30 ABSTRACT:
0000 31 This module contains the routine that invokes a COBOL USE procedure.
0000 32
0000 33
0000 34 --
0000 35
0000 36 VERSION: 1
0000 37
0000 38 HISTORY:
0000 39
0000 40 AUTHOR:
0000 41 Marty Jack, 2-May-1979
0000 42
0000 43 MODIFIED BY:
0000 44
0000 45
0000 46
```



COB\$\$\$INVOKE\_USE  
1-004

COBOL Invoke USE Procedure B 8 15-SEP-1984 23:45:30 VAX/VMS Macro V04-00 Page 2  
HISTORY ; Detailed Current Edit History 6-SEP-1984 10:46:58 [COBRTL.SRC]COBINVUSE.MAR;1 (2)

```
0000 48 .SBTTL HISTORY ; Detailed Current Edit History
0000 49
0000 50
0000 51 : Edit History for Version 1 of COBINVUSE
0000 52 :
0000 53 : 1-001 - Original. MLJ 2-May-1979
0000 54 : 1-002 - Change entry point name to COB$$$INVOKE_USE
0000 55 : 1-003 - Signal 'Recursive activation' directly. Cosmetic changes.
0000 56 : RKR 20-OCT-79
0000 57 : 1-004 - Signalling should be 'Recursive activation of USE procedure'
0000 58 : RKR 29-OCT-79
```

```
0000 60 .SBTTL DECLARATIONS
0000 61
0000 62 :
0000 63 : INCLUDE FILES:
0000 64 : $SFDEF
0000 65
0000 66 :
0000 67 : EXTERNAL SYMBOLS:
0000 68 : .DISABLE GBL
0000 69 : .EXTRN LIB$STOP : Signal a condition and stop
0000 70 : .EXTRN COB$HANDLER : COBOL exception handler
0000 71 : .EXTRN COB$USE_EXIT : Special COBOL exception for USE EXIT
0000 72 : .EXTRN COB$REACTUSE : Recursive activation of USE procedure
0000 73 : .EXTRN LIB$SIGNAL : Signal a condition
0000 74 : .EXTRN LIB$STOP : Signal a condition and stop
0000 75 : .EXTRN OTSS_FATINTERR : Fatal internal error
0000 76
0000 77 :
0000 78 : MACROS:
0000 79 : : NONE
0000 80 :
0000 81 :
0000 82 :
0000 83 : PSECT DECLARATIONS:
0000 84 : .PSECT _COB$CODE PIC, SHR, LONG, EXE, NOWRT
0000 85
0000 86 :
0000 87 : EQUATED SYMBOLS:
0000 88 : :
0000 89 : : NONE
0000 90 :
0000 91 : OWN STORAGE:
0000 92 : : NONE
0000 93 :
```

```
0000 95 .SBTTL COB$$$INVOKE_USE
0000 96
0000 97 :++
0000 98 : FUNCTIONAL DESCRIPTION:
0000 99 :
0000 100 : This routine invokes a COBOL USE procedure.
0000 101 :
0000 102 : CALLING SEQUENCE:
0000 103 :
0000 104 : COB$$$INVOKE_USE (proc.rl.v, uselist.rl.v, apval.rl.v, eopr.mq.r, pnc.ml.r)
0000 105 :
0000 106 : INPUT PARAMETERS:
0000 107 :
0000 108 : PROC.rl.v The address of the USE procedure
0000 109 : USELIST.rl.v The address of the USE list of the program
0000 110 : in which the original exception was detected
0000 111 : APVAL.rl.v The AP of the program in which the original
0000 112 : exception was detected
0000 113 :
0000 114 : IMPLICIT INPUTS:
0000 115 :
0000 116 : NONE
0000 117 :
0000 118 : OUTPUT PARAMETERS:
0000 119 :
0000 120 : EOPR.mq.r The end of perform range block for the USE
0000 121 : procedure to be invoked.
0000 122 : PNC.ml.r The perform nest counter for the module in
0000 123 : which the USE procedure is contained.
0000 124 :
0000 125 : IMPLICIT OUTPUTS:
0000 126 :
0000 127 : NONE
0000 128 :
0000 129 : FUNCTION VALUE:
0000 130 :
0000 131 : NONE
0000 132 :
0000 133 : SIDE EFFECTS:
0000 134 :
0000 135 : The USE procedure is executed.
0000 136 :
0000 137 :--
0000 138 :
0000 139 :
OFFC 0000 140 .ENTRY COB$$$INVOKE_USE,-
0002 141 ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0002 142 : All registers MUST be saved since
0002 143 : the USE procedure will not.
0002 144 :
00000004 0002 145 proc= 4
00000008 0002 146 uselist=8
0000000C 0002 147 apval= 12
00000010 0002 148 eopr= 16
00000014 0002 149 pnc= 20
0002 150
0002 151 :+
```



```
0002 152 : Load most parameters into registers. The USE-list pointer goes to -4(FP)
0002 153 : and the AP value goes to AP. Note that after this point, the argument
0002 154 : list cannot be accessed, since AP has been modified. The effect of this
0002 155 : code is that the USE procedure will be executed with the same AP and the
0002 156 : same USE-list as the routine that raised the exception originally.
0002 157 :-
50 10 AC D0 0002 158 : MOVL eopr(AP),R0 ; Address of end of perform block
51 14 AC D0 0006 159 : MOVL pnc(AP),R1 ; Address of perform nest counter
52 04 AC D0 000A 160 : MOVL proc(AP),R2 ; Address of USE procedure
5C 08 AC DD 000E 161 : PUSHL uselist(AP) ; USE-list address to -4(FP)
5C 0C AC D0 0011 162 : MOVL apval(AP),AP ; AP value to AP
0015 163 :-
0015 164 :+
0015 165 : Ensure that the USE procedure is inactive. If it is active, signal an
0015 166 : error.
0015 167 :-
60 0D D5 0015 168 : TSTL (R0) ; Test for nonzero return address
0D 13 0017 169 : BEQL 1$ ; Br if inactive
00000000'8F DD 0019 170 : PUSHL #COB$ RECACTUSE ; Recursive activation of USE procedure
00000000'GF 01 FB 001F 171 : CALLS #1, G^LIB$STOP ;
0026 172 :-
0026 173 :+
0026 174 : Establish COB$HANDLER as the handler. This is necessary so that (1) it will
0026 175 : get control if an exception occurs during execution of the USE procedure and
0026 176 : (2) so that COB$IOEXCEPTION will recognize that -4(FP) contains a valid
0026 177 : USE-list pointer in case the USE procedure calls it (recursively).
0026 178 :-
6D 00000000'GF 9E 0026 179 1$: MOVAB G^COB$HANDLER,(FP)
002D 180 :-
002D 181 :+
002D 182 : Save the return PC from the frame and overwrite it with the address of a
002D 183 : handler that will get control if the USE procedure executes an EXIT PROGRAM
002D 184 : (that is, a RET instruction).
002D 185 :-
10 AD 10 AD DD 002D 186 : PUSHL SF$ SAVE_PC(FP) ; Push return PC on stack
46'AF 9E 0030 187 : MOVAB B^10$,SF$ SAVE_PC(FP) ; Overwrite with handler address
0035 188 :-
0035 189 :+
0035 190 : Invoke the USE procedure using the PERFORM invocation code.
0035 191 :-
04 A0 61 D7 0035 192 : DECL (R1) ; Decrement the perform nest counter
60 41'AF 9E 0037 193 : MOVL (R1),4(R0) ; Move it to EOPR block
62 17 003B 194 : MOVAB B^2$, (R0) ; Move exit address to EOPR block
003F 195 : JMP (R2) ; Go to USE procedure
0041 196 :-
0041 197 :+
0041 198 : If control reaches this point, the USE procedure terminated by executing
0041 199 : the end of perform range. Restore the return address, and execute a RET
0041 200 : to return to COB$IOEXCEPTION.
0041 201 :-
10 AD 8E D0 0041 202 2$: MOVL (SP)+,SF$ SAVE_PC(FP) ; Restore return address
04 04 0045 203 : RET ; Return
0046 204 :-
0046 205 :+
0046 206 : If control reaches this point, the USE procedure terminated by executing
0046 207 : an EXIT PROGRAM. (Note also that this removed the stack frame that we were
0046 208 : executing in, and we are now executing in the frame of COB$IOEXCEPTION.)
```



COB\$\$\$INVOKE\_USE  
1-004

COBOL Invoke USE Procedure  
COB\$\$\$INVOKE\_USE

F 8

15-SEP-1984 23:45:30 VAX/VMS Macro V04-00 Page 6  
6-SEP-1984 10:46:58 [COBRTL.SRC]COBINVUSE.MAR;1 (4)

```
00000000'8F DD 0046 209 : We must cause the program that originally invoked COB$IOEXCEPTION to return.
00000000'GF 01 FB 0046 210 : We do this by signalling a condition that is intercepted by the COB$HANDLER
00000000'8F DD 0046 211 : that is declared in that frame, which will call SYSSUNWIND.
00000000'GF 01 FB 0046 212 :
0046 213 10$: PUSHL #COB$ USE EXIT ; Special exception condition
004C 214 CALLS #1, G*LIB$SIGNAL ; Signal it
0053 215 PUSHL #OFSS FATINTERR ; If didn't unwind, fatal error
0059 216 CALLS #1, G*LIB$STOP ;
0060 217 :
0060 218 .END
```

COB\$\$\$INVOKE\_USE  
Symbol table

COBOL Invoke USE Procedure

G 8

15-SEP-1984 23:45:30 VAX/VMS Macro V04-00  
6-SEP-1984 10:46:58 [COBRTL.SRC]COBINVUSE.MAR;1

Page 7  
(4)

APVAL = 0000000C  
COB\$\$\$INVOKE\_USE 00000000 RG 02  
COB\$HANDLER \*\*\*\*\* X 00  
COB\$REACTUSE \*\*\*\*\* X 00  
COB\$USE\_EXIT \*\*\*\*\* X 00  
EOPR = 00000010  
LIB\$SIGNAL \*\*\*\*\* X 00  
LIB\$STOP \*\*\*\*\* X 00  
OT\$FATINTERR \*\*\*\*\* X 00  
PNC = 00000014  
PROC = 00000004  
SF\$SAVE\_PC = 00000010  
USELIST = 00000008

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes															
ABS	00000000 ( 0.)	00 ( 0.)	NOPI	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
\$ABS\$	00000000 ( 0.)	01 ( 1.)	NOPI	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					
_COB\$CODE	00000060 ( 96.)	02 ( 2.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG					

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.03	00:00:00.96
Command processing	115	00:00:00.32	00:00:02.70
Pass 1	112	00:00:00.69	00:00:05.60
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	54	00:00:00.29	00:00:02.05
Symbol table output	3	00:00:00.01	00:00:00.13
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	318	00:00:01.37	00:00:11.48

The working set limit was 1050 pages.  
4095 bytes (8 pages) of virtual memory were used to buffer the intermediate code.  
There were 10 pages of symbol table space allocated to hold 41 non-local and 3 local symbols.  
218 source lines were read in Pass 1, producing 13 object records in Pass 2.  
8 pages of virtual memory were used to define 7 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4

88 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

COB\$\$\$INVOKE USE  
VAX-11 Macro Run Statistics

COBOL Invoke USE Procedure

H 8

15-SEP-1984 23:45:30 VAX/VMS Macro V04-00 Page 8  
6-SEP-1984 10:46:58 [COBRTL.SRC]COBINVUSE.MAR;1 (4)

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:COBINVUSE/OBJ=OBJ\$:COBINVUSE MSRC\$:COBINVUSE/UPDATE=(ENH\$:COBINVUSE)



0063

AH-BT13A-SE  
VAX/VMS V4.0

**DIGITAL EQUIPMENT CORPORATION**  
**CONFIDENTIAL AND PROPRIETARY**